

Maintenance Planning Guideline

Use this guideline to plan for work that happens after releasing a product or system to customers, as well as an annotated outline for writing a *Maintenance Plan*. This guideline includes an introduction to what maintenance involves, maintenance types, and how maintenance planning fits within the system development or product to be maintained.

Although the development project may be completed and released, the project deliverable simply enters a new phase of its lifecycle. Whether hardware, software, or system, the major project deliverables require ongoing support from the company. That support includes such activities as responding to user issues, maintaining the system hardware, incorporating enhancements, and performing preventative maintenance. For the success of the product or system in the field, thoroughly planning ahead for proper maintenance support is crucial.

1. Read the guidelines starting on page 2:
 - Introduction to Maintenance
 - Defining the “Maintenance Concept”
 - Maintenance Planning Overview—Types and Timing
2. Determine which aspects of maintenance apply to the project deliverables.
3. Using the information on page 3, “Maintenance Planning Overview—Types and Timing,” determine when you should begin writing a maintenance plan in your project cycle.
4. Adapt the Maintenance Plan outline (starting on page 5) to your specific projects.
5. If your project is in its early stages, add maintenance planning activities to your project schedule.
6. If your project has already started and there is no maintenance plan, have a maintenance planning meeting so discussions and planning information can benefit any associated design work.
7. Ensure that maintenance planning includes a specific “owner” going forward, who continues to be a regular part of your project, and has adequate inputs to development and test reviews.
8. As you move through the project, update the plan iteratively; thus, yielding a detailed maintenance plan later in the project to ensure all aspects of personnel and processes are in place when the product or system is released to production/deployment.

Maintenance Planning Guidelines and Plan Outline

Introduction to Maintenance

Maintenance refers to ongoing support and evolution of a system once it is released to users/customers. Maintenance planning needs to occur while a system is in the development stage.

Purpose of the Maintenance Process:

- Sustain and monitor system capability to provide services.
- Record problems for analysis.
- Take corrective, adaptive, perfective, and preventive actions.
- Confirm restoration capability.

Expected Outcomes of Maintenance Process:

- A maintenance strategy is developed.
- Maintenance constraints are provided as inputs to requirements.
- Replacement system elements are made available.
- Services meeting stakeholder requirements are sustained.
- The need for corrective design changes is reported.
- Failure and lifetime data is recorded.

From ISO/IEC 15288:2002, copyright ISO/IEC 2002

Defining the Maintenance Concept

The maintenance concept is developed first. It is really the high-level design of the approach to system maintenance. This concept sets the overall parameters for doing more detailed maintenance planning. It should cover:

- **The scope of the maintenance**, e.g. an overview of covered hardware and software systems/subsystems; whether maintenance of third-party system elements is included; what types of maintenance are covered (see list below).
- **Post-delivery processes**, e.g. an overview of the approach to problem resolution, change control, and deployment of updates; whether any tailoring from standard company processes is necessary or allowable.

- **Overall responsibilities for maintenance**, what group(s) will be responsible for fulfilling which aspects of the maintenance processes.
- **Estimates of life-cycle costs**, guidelines for acceptable life-cycle costs that will drive maintenance approaches and budgets.

Based on the overall guidelines established in this maintenance concept, a detailed maintenance plan can be developed.

Maintenance Planning Overview: Types and Timing

The Maintenance Plan must address detailed processes for the development of changes as well as their deployment. It addresses subjects such as:

- Scope of maintenance effort
- Schedule for any regular maintenance releases
- Types of maintenance releases allowed
- Maintenance processes and techniques, including for issues recording, escalation, prioritization, and change reviews and control
- Organizational responsibilities and staffing
- Tools and equipment including configuration management, defect tracking, spares requirements, and test equipment
- Processes for acceptance testing, releases, and deployment communication
- Budgets
- Ongoing monitoring of maintenance effectiveness and customer satisfaction with the system

Major aspects of maintenance which must be addressed in maintenance planning:

- Maintaining control over the system's day-to-day functions
- Maintaining control over system modification
- Perfecting existing acceptable functions
- Preventing system performance from degrading to unacceptable levels

Categories of maintenance:

- **Corrective Maintenance** – Changes necessitated by actual errors/bugs or design deficiencies. Corrective maintenance consists of activities normally

considered to be error correction required to keep the system operational. By its nature, corrective maintenance is usually a reactive process. Corrective maintenance is related to the system not performing as originally intended. The three main causes of corrective maintenance are (1) design errors, (2) logic errors, and (3) coding errors.

- **Adaptive Maintenance** – Changes initiated as a result of changes in the environment in which a system must operate. These environmental changes are normally beyond the control of the maintainer and consist primarily of changes to the: (1) rules, laws, and regulations that affect the system; (2) hardware configuration, e.g., new terminals, local printers, etc.; (3) data formats and file structures; and (4) system software, e.g., operating systems, compilers, utilities, etc.
- **Perfective Maintenance** (also known as enhancements and upgrades) – All changes, insertions, deletions, modifications, extensions, and enhancements made to a system to meet the evolving and/or expanding needs of the user. It is generally performed as a result of new or changing requirements, or in an attempt to augment or fine-tune the existing software/hardware operations/performance. Activities designed to make the code easier to understand and to work with, such as restructuring or documentation updates are included in the Perfective category.
- **Preventive Maintenance** – Changes required to avoid problems or detect problems before they cause operational problems. Hardware maintenance could include regular checks of performance and physical inspection for wear and resulting adjustments as necessary. Software maintenance in this category could include regular review of performance metrics, analysis of loads and trends and any emerging issues with system performance, and adjustments of the system to ensure that operations are not disrupted.

How Maintenance Planning and Preparation Fit in the System Development Life Cycle

An inherent element of the system requirements is the maintainability of the system. This can be captured in the design alternatives analysis, maintained as part of the configuration management of the system, and verified during the operational phase of the project. The maintenance concept above leads to high-level and detailed maintenance requirements. Implementation of the system would also result in implementation of the maintenance management program, verification that the system is as maintainable as hoped, and finally concurrent operations and maintenance activities. The cross-cutting activity of system validation assessment during the operation phase would also be paralleled by a validation that the maintenance concept was captured in the management of the system and the verification that maintenance requirements were being met.

Taking it phase by phase, consideration of maintenance implications leads to this typical involvement of support personnel during development of the system.

- System requirements definition and alternatives investigation:**
 Maintenance concept development and planning must be done in parallel with the development of the system to be maintained, starting with overall requirements definition and investigation of design alternatives. The overall concept of maintenance may drive tradeoffs in system development. For example, how much time and money will be spent developing highly automated diagnostic tools to ease troubleshooting and contain costs of field visits? Or developing sophisticated online automated code update capabilities to reduce physical shipment of new software versions to the field? Those who will be responsible for maintenance should be involved in such trade-off discussions early in the project so that the life-cycle costs associated with maintenance are considered in the initial system design.
- Development:** As elements of the system are developed (detailed design and implementation performed), features that influence or implement directly the maintenance concept are created by the development team. Those responsible for maintenance and support should be included in appropriate reviews and prototyping. They will have detailed input to how to make the system most easily diagnosed, maintainability of sub-systems, etc.
- Testing and documentation:** As the time for deployment nears, maintenance personnel can begin to learn a new system, even assisting with testing to learn hands-on about installation and configuration and system troubleshooting. They can also review and help hone technical publications such as user manuals and installation guides.
- Deployment:** System deployment marks the official beginning of transition to ongoing maintenance and support. The handoff between development and maintenance personnel should be done with attention to understanding of state of the system—open issues, any features that did not make the release, etc., as well as readiness of the maintenance organization and processes and any need for initial support from development.

MAINTENANCE PLAN OUTLINE:

Typical Items Covered in a Maintenance Plan

1. Overview and Scope

1. Types of Maintenance included

Provide brief overview statements of how/whether corrective, adaptive, preventive, and perfective maintenance is covered by this plan.

2. Systems Included in Software Maintenance

Calls out the types of software included in the maintenance and corresponding systems or subsystems, such as:

- **Operating System**: Maintenance issues include system administration and network security, upgrades or “patches” to the latest version and dealing with potential incompatibilities between the OS and other hardware and software in the system.
- **Commercial Off-the-Shelf**: Sometimes called COTS software, this includes commercially available applications that are part of the overall operation, such as relational database management systems, word processing, and/or spreadsheets. Examples include Oracle, Word, and Excel. Maintenance issues include configuration and user administration of COTS implementations, software maintenance contracts, upgrades, and potential incompatibilities between COTS and other hardware and software in the system.
- **Application Software**: Maintenance issues include system configuration management, software maintenance contracts, backup and disaster recovery, user and operator administration, and system security. In addition, the issue of intellectual property rights and source code ownership has a tremendous impact on maintenance issues and costs.

3. Items Included in Hardware Maintenance

Calls out all hardware (electronic, mechanical) elements of the product/system that must be maintained, including enclosures, electronics, and third-party modules or systems.

4. Release and Maintenance Phase Schedule

Provides a timeline view showing key milestones and release cycles, including the following items:

- **Initial deployment**: The schedule should show the start of ongoing maintenance with respect to initial deployment—for example, is there a

transition period during which Development or other non-maintenance personnel are expected to stay involved or continue to lead efforts? The schedule should define at which point each responsible group is assuming ownership of support.

- **Regularly-scheduled/ongoing maintenance**: Summarize and provide an overview timeline of the overall maintenance cycle. For example, for software, show the schedule for the types of maintenance releases covered by this plan. Also, hardware-related timelines can show regular preventive maintenance trips. Software-related timelines can show regular back-up schedules and database maintenance. See next section, Types of Maintenance Releases. All types appropriate for your company should show up in the Maintenance Schedule, with either hard timing or estimates of frequency.
- **Ad hoc individual patches**: e.g. small, timely fixes for urgent problems.
- **Regular patch releases**: e.g. larger subsystem roll-up patches for the delivery of less time critical fixes.
- **Full system deliveries**: Driven by integration activities that require the establishment of a new maintenance baseline.
- **External milestones**: For example, releases of new versions of operating systems or third party components that must be migrated into the system.
- **Linkage with company development plans**: By definition maintenance involves supporting evolving systems. The linkage of maintenance activities to ongoing development programs should be shown, i.e. is there a cycle by which maintenance items are fed into development portfolio planning?

5. Types of Maintenance Releases

Some companies will define different levels of release allowable under various circumstances; specifically, the ability to quickly release engineering or test level code to quickly address emergency situations. If such cases are to be allowed, the maintenance plan can specify constraints and guidelines for each type of release. The following paragraphs are examples of software releases:

- **Engineering Software**
 - **Definition and purpose**: Engineering software is software delivered in response to an emergency. It is only provided at the specific request of an internal group, with the goal of mitigating a critical operations problem within hours of the onset of the problem.
 - **Approach**: Engineering software may be delivered at any time of the day or day of the week. Engineering software is built in a developer's view from source code that is not yet merged to the baseline and typically only unit tested by the developer. The goal within sustaining

engineering will be to merge corresponding fixes (tested and approved) to the appropriate baseline within 48 hours of sending the engineering software. The team will continue to use a custom ClearCase tool developed that controls and documents the delivery of engineering software to the field. This tool, accessible only by senior developers (subsystem leads) who are authorized to send engineering software, captures and documents what engineering software has been sent.

- Test Executables (TEs)
 - Definition and purpose: Test executables are software deliveries designed to fix a specific problem with the smallest possible delivery footprint. The footprint of a change is the number of delivered components required to implement the change. The goal of a TE is to provide a fix for an urgent problem as soon as it is available (possibly before final, definitive testing is completed), in a form that can be promoted into operations by the internal receiving group as quickly as possible. TEs are also only provided at the specific request of a user location. TEs are normally delivered during the standard work week, but can be built and delivered during off hours for especially urgent problems.
 - Approach: TEs are delivered from the controlled baseline (after a merge and build), either from the maintenance baseline or from an Emergency Bug Fix branch. TEs are delivered under a configuration change request routed through the change control board. Software CM uses custom tools to send the TE tar files, document the receipt of the delivery by each site, and distribute all of the technical data to all interested parties. TEs are installed and tested by the test group in the test environment, but this testing may be done in parallel with the delivery to the user site depending on the urgency of the request. The delivery of TEs will be planned and managed on a daily basis by the lead of the Deployment team. The Custom Software Delivery process governs the delivery of Test Executables as well as Patches.
- Patches
 - Definition and purpose: Patches are larger software deliveries, usually covering an entire subsystem. The goal of a patch is to deliver all of the fixes that have been applied to the maintenance baseline for an entire component or subsystem. Sometimes the fix to a problem will require the simultaneous delivery and installation of components from multiple subsystems; when this happens, multiple subsystem patches are generated, tested, and delivered as a group. Patch deliveries are planned and scheduled by the Deployment team, taking into account the program priority list, the list of program mission milestones, development progress in merging fixes for specific problems, and inter-relationships between fixes among

subsystems. We will provide a subsystem patch for each major subsystem every 6 to 8 weeks.

- Approach: Patch delivery follows a rigorous process. The Software Integration and Test Team sends a request to software CM to build tar files for the patch, identifies the issues fixed in the patch, and generates draft installation instructions. The test group tests the installation of the patch in the code control system, providing redlines to the installation instructions as required, and verifies the issues fixed in the patch. Additional functional regression testing is performed as required. When testing is completed, the patch is presented at a pre-ship review by the DPT. The installation instructions and the issues are discussed with the receiving user sites, including any changes in operational procedures or troubleshooting methods required by the fixes. A change control record is executed to deliver the software, and software CM uses custom tools to send the patch tar files, document the receipt of the delivery by each site, and distribute the technical data package for the delivery.

- Drop/Release

The delivery of the full system is usually referred to as a drop or a release, and is accompanied by the transition to a new maintenance baseline. Full system deliveries will be performed only as necessary to deliver capabilities and upgrades that require changes in a majority of the subsystems. Extensive regression and performance testing precede full system deliveries.

- Transition/Training

Occasionally, deliveries will require a set of transition activities to be performed (e.g. building of a database index or updating values in a database table) either before or after installation of the software. Scripts are provided to perform these activities. A detailed transition plan is developed and the users are provided hands-on transition training. With proactive support from sustaining engineers, the customer will test the installation. Feedback from this exercise is incorporated into installation and transition instructions prior to deployment to all locations. For new capabilities not requiring the scope of support described above, the primary development engineer is invited to one of the bi-weekly deployment team meetings to discuss the new capability.

2. Maintenance Staffing and Environment

1. Staffing and Roles and Responsibilities

- **Overall responsibilities and required skills and resources:** This section makes clear who has responsibility for what maintenance activities, and what resources in the support organization(s) are involved (at least by skill set and number of resources, if not by name).

- **Leads and customer contacts:** This section identifies key leads, as well as any primary contacts for user groups or customer sites. Those customer/user contacts should include specific contacts for different maintenance activities, e.g. the contact for escalation of user issues, for planning cut-in of new releases, etc.

2. Tools and Equipment Needs for Maintenance Activities

- **“Infrastructure” inherent to the Maintenance process:** Internal tools such as configuration management software and defect and enhancement request tracking databases.
- **Spare Parts:** Specify inventory of spare parts to replace failed or damaged equipment during ongoing operations and maintenance. Many maintenance contracts and even some procurement contracts use a rule of thumb of between 5 and 10 percent spares; that is, 5 to 10 percent times the total installed base of certain critical items. A quantitative estimate for spare parts may be calculated if certain variables are known or can be estimated with confidence. For example, if the MTBF and MTRR are known, it is possible to calculate how many spares will be required to have on hand to meet a desired level of availability (i.e. 95% of all cameras online at any given time).
- **Test Equipment and Test Beds:** Specify test equipment or environments that must be available to support debugging of problems and testing of changes before they are released to production.

3. Training Required

- **Maintenance/support personnel:** Define what training the company’s support personnel should receive (usually prior to release) in maintenance procedures.
- **User/Customer training:** Define what training the users of the product or system should receive related to their role in maintaining the system (for example, proper back-ups).

3. Maintenance Procedures

1. Handling change requests

The following procedures must be developed and documented in the Maintenance Plan (or the Maintenance Plan would reference company processes for these items).

- Fielding change requests
 - Submitting modification requests
 - Reviewing and prioritizing requests
- Developing and testing changes

- Analyzing and verifying a problem, developing implementation options
- Performing reviews of implementation options, assessing system and company impacts, and choosing an option
- Performing technical reviews of changes and approving those changes
- Implementing and testing the changes
- Release planning and configuration management
 - Allocating problems and fixes to specific releases
 - Defining deployment of changes, including developing and communicating migration plans
 - Implementing configuration management of the system and changes to the system
 - Deciding on retirement of particular elements of a system, planning and communicating the retirement, and executing switchover

2. Acceptance Testing and Signoff

The plan can indicate what acceptance testing processes will be employed and signoffs required for different types of releases.

This might include referencing specific User Acceptance Test plans that must be run or Customer Acceptance Checklists that must be executed by sites accepting new hardware or software releases.

3. Deployment Communication

Delivery plans for software updates need to be flexible and clearly communicated. The Maintenance Plan can identify what approaches will be used to communicate deployment of system changes, such as:

- **Teleconferences**: Releases can have a standard release communication teleconference scheduled, where representatives from affected groups can hear a summary of the upcoming release and deployment plans and ask questions about how their environment is affected.
- **Email**: Notification to a standard distribution list of affected groups as well as other parties who need to stay informed about the big picture, i.e. to understand how often the system is being updated, with what kinds of changes. Both delivery plans and distribution notices should be communicated.
- **“Patch plans”**: The maintenance organization can generate and distribute a regular patch plan communicating a look-ahead schedule for all deliveries to the field. For example, it can document the 3-month plan; this plan should

be updated more frequently than the look-ahead period, to keep it current with any changes in priorities.

4. Life-cycle Costs and Maintenance Budgets

Based on the staffing, equipment needs, and processes to be included in maintenance, this section summarizes expected life-cycle costs for the product or system and outlines the budget requirements for executing the maintenance.

5. System/Product Design Implications from the Maintenance Concept and Processes

As this plan is developed, the team is thinking through how the system will be maintained and supported. This planning process will provide insights that must be communicated to the design team. For example, the maintenance concept and process details may require that specific diagnostic tools be created or certain error codes be included in the software.

6. Performance Monitoring and Management during the Maintenance Phase

1. Measures of Performance

Whether the maintenance activities are conducted by in-house personnel or contracted out, there are several measures of performance that can be used to determine the state of the original system when released, as well as the effectiveness of ongoing maintenance. These include:

- **Mean time between failures** – the average time between device failures, usually expressed in hours.
- **Mean time to repair** – the average time to repair (or replace) a device, typically this includes the response time, expressed in hours.
- **System Availability** – the time that the system provides its designed functionality, expressed in hours. Typically, this excludes scheduled downtime due to maintenance or system administration activities.
- **System Reliability** – similar to Availability, but expressed as the probability that the system will be available.
- **User help-desk inquiries** – measure of how much support users required.
- **Defects reported** – measure of how many problems were discovered after release.

2. Monitoring of Customer Satisfaction – Regular Reviews

This section documents how customer satisfaction with system performance will be assessed as part of the maintenance process. The goal is to have up-to-date understanding of the needs of the customer and the perceived performance of

the current system, and to identify possible future applications to incorporate into future releases. Although it is expected that users will raise issues through the issues reporting or enhancement request system, the goal of this section is to also identify a regular overarching review process with the users—periodically scheduling a time to review status and solicit input for improvements.

This section of the plan should include:

- How the measures of performance above will be measured at customers
- Frequency of reviewing enhancement request database
- Frequency and types of regular reviews with customers, including not only system functionality but also any new needs for training or documentation
- How issues or requests raised in these forums will be prioritized and assigned, and how results communicated to the customer

Note: this is often a sales and marketing function, since those groups need to feed customer requests back into their product or system development plans. But since the support teams are often the closest to the customer, they can lead or be involved in this feedback process.

3. Performance Reports

This section calls out what reports should be produced regularly as part of the above processes, what they should contain, and who should receive them.

4. Managing Execution of the Maintenance Plan

With the maintenance plan developed, approved, and funded, there must be a structured practice for managing the plan. The basics of plan management are similar to other practices and should include:

- **Performance Monitoring:** Regular checking of plan metrics and budgets against the maintenance plan projections, as documented in 5.1 above
- **Oversight Support:** Support of the plan over time and its relationship to training, and staffing issues
- **On-going Multi-year Planning:** Plan for changes in system components, emerging issues, changes in the process and new or evolving needs of stakeholders. Check that the level of effort and resources are appropriate to support a multi-year maintenance program plan.
- **Operational Needs:** Check that the maintenance concept is consistent with operational concept.

The Maintenance Plan document can specify how each of the above will be addressed.

Appendices

Possible references or appendices to the Maintenance Plan:

- Related company processes documents – configuration management, escalation, etc.
- Forms to be used during maintenance (e.g. change request, software release, etc.)

Administrative Information

Revision	Author	Date	Sections Affected	Change Summary
1.0		1/3/2009		

Current Version	1.0
Date	1/3/2009
Master Document Chapter Number	1
Document ID	20